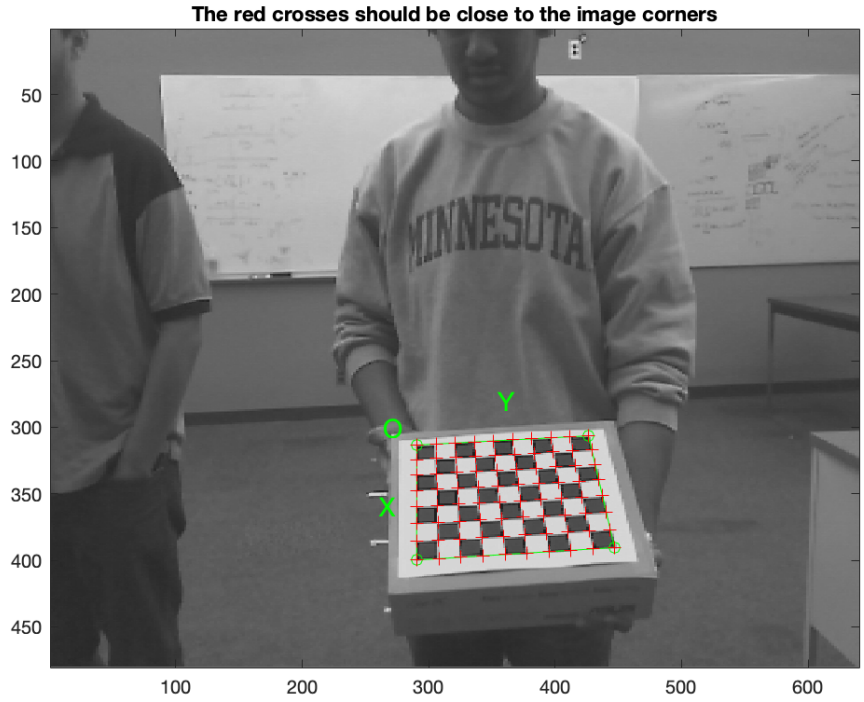
CSE 6367 Assignment #5

Zahidur Talukder

April 13, 2021

Problem 1

In this Problem, we have to use the MATLAB camera calibration toolbox to extract the intrinsic and extrinsic parameters for the stereo camera. I have used the documentation provided for this task and calibrate the left and right camera separately first using **"calib_gui.m"** and after adding the images and extracting grid corners and using calibration we get 4 files, names **"Calib_Results_left.mat"**, **"Calib_Results_right.mat"**, **"Calib_Results_left.m"**, **"Calib_Results_right.m"** with the intrinsic and extrinsic parameter for individual cameras. Then using **"stereo_gui.m"** we load again the **"Load left and right calibration file"** and after running **"Run stereo calibration"** we finally get the intrinsic and extrinsic parameter for both camera as **"Calib_Results_stereo.mat"**. I am attaching all the file in the report as zip file and give the screenshot of the parameter I got.

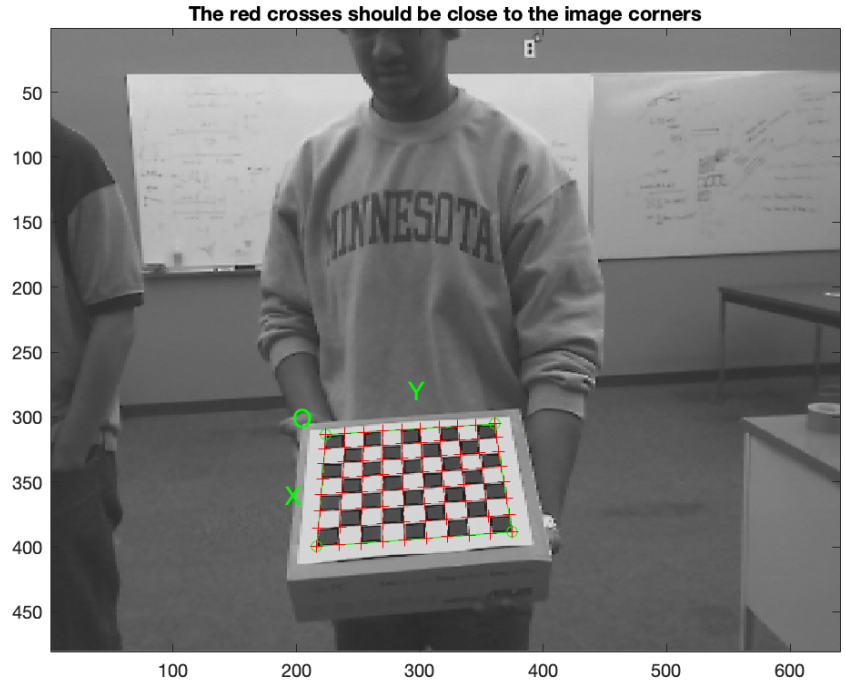


(a) Showing Extracting Grid Corner for Calibration for 1 Image out of 12

```
Calibration parameters after initialization:

Focal Length:          fc = [ 646.76015   646.76015 ]
Principal point:       cc = [ 319.50000   239.50000 ]
Skew:             alpha_c = [ 0.00000 ]   => angle of pixel = 90.00000 degrees
Distortion:            kc = [ 0.00000   0.00000   0.00000   0.00000   0.00000 ]

Main calibration optimization procedure - Number of images: 12
Gradient descent iterations: 1...2...3...4...5...6...7...8...9...10...11...12...13...14...15...16...17...18...19...20...21...22...23...
Estimation of uncertainties...done


Calibration results after optimization (with uncertainties):

Focal Length:          fc = [ 672.83898   666.86302 ] +/- [ 13.76200   12.77369 ]
Principal point:       cc = [ 376.79465   242.88516 ] +/- [ 17.63474   11.62596 ]
Skew:             alpha_c = [ 0.00000 ] +/- [ 0.00000  ]   => angle of pixel axes = 90.00000 +/- 0.00000 degrees
Distortion:            kc = [ -0.00995   0.44666   -0.00182   0.02040  0.00000 ] +/- [ 0.07975   0.44057   0.00715   0.01104  0.00000 ]
Pixel error:          err = [ 0.46455   0.39443 ]

Note: The numerical errors are approximately three times the standard deviations (for reference).


Recommendation: Some distortion coefficients are found equal to zero (within their uncertainties).
                To reject them from the optimization set est_dist=[0;1;1;1;0] and run Calibration
```

(b) Intrinsic Parameters for Individual Left camera

Figure 1: Calibarating Left Camera to get Camera Parameters



(a) Showing Extracting Grid Corner for Calibration for 1 Image out of 12

```
Calibration results after optimization (with uncertainties):

Focal Length:          fc = [ 667.49720   664.07127 ] +/- [ 12.29849   11.18697 ]
Principal point:       cc = [ 353.93918   217.44607 ] +/- [ 16.77757   13.49686 ]
Skew:             alpha_c = [ 0.00000 ] +/- [ 0.00000  ]   => angle of pixel axes = 90.00000 +/- 0.00000 degrees
Distortion:            kc = [ 0.05427   -0.06253   -0.00955   0.00830  0.00000 ] +/- [ 0.06491   0.17037   0.00681
Pixel error:          err = [ 0.43676   0.37306 ]

Note: The numerical errors are approximately three times the standard deviations (for reference).
```

(b) Intrinsic Parameters for Individual Right camera

Figure 2: Calibarating Right Camera to get Camera Parameters

```
Stereo calibration parameters after loading the individual calibration files:


Intrinsic parameters of left camera:

Focal Length:          fc_left = [ 672.83898   666.86302 ] ± [ 13.76200   12.77369 ]
Principal point:       cc_left = [ 376.79465   242.88516 ] ± [ 17.63474   11.62596 ]
Skew:             alpha_c_left = [ 0.00000 ] ± [ 0.00000  ]   => angle of pixel axes = 90.00000 ± 0.00000 degrees
Distortion:            kc_left = [ -0.00995   0.44666   -0.00182   0.02040  0.00000 ] ± [ 0.07975   0.44057   0.00715   0.01104  0.00000 ]


Intrinsic parameters of right camera:

Focal Length:          fc_right = [ 667.49720   664.07127 ] ± [ 12.29849   11.18697 ]
Principal point:       cc_right = [ 353.93918   217.44607 ] ± [ 16.77757   13.49686 ]
Skew:             alpha_c_right = [ 0.00000 ] ± [ 0.00000  ]   => angle of pixel axes = 90.00000 ± 0.00000 degrees
Distortion:            kc_right = [ 0.05427   -0.06253   -0.00955   0.00830  0.00000 ] ± [ 0.06491   0.17037   0.00681   0.01126  0.00000 ]


Extrinsic parameters (position of right camera wrt left camera):

Rotation vector:             om = [ -0.03954   0.00959  -0.01218 ]
Translation vector:           T = [ -83.11051   -2.08717  -1.00017 ]
```

(a) Intrinsic and Extrinsic Parameter for Stereo Camera after Loading both camera files

```
Stereo calibration parameters after optimization:


Intrinsic parameters of left camera:

Focal Length:          fc_left = [ 693.41314   671.10016 ] ± [ 29.02352   23.93891 ]
Principal point:       cc_left = [ 438.45401   241.44020 ] ± [ 33.67967   10.30011 ]
Skew:             alpha_c_left = [ 0.00000 ] ± [ 0.00000  ]   => angle of pixel axes = 90.00000 ± 0.00000 degrees
Distortion:            kc_left = [ -0.30966   1.13193   0.00092   0.03139  0.00000 ] ± [ 0.13253   0.76201   0.00574   0.01462  0.00000 ]


Intrinsic parameters of right camera:

Focal Length:          fc_right = [ 642.04990   653.18864 ] ± [ 24.90820   22.78903 ]
Principal point:       cc_right = [ 442.10875   210.51757 ] ± [ 34.04651   23.11566 ]
Skew:             alpha_c_right = [ 0.00000 ] ± [ 0.00000  ]   => angle of pixel axes = 90.00000 ± 0.00000 degrees
Distortion:            kc_right = [ -0.12272   -0.04172   -0.00306   -0.01835  0.00000 ] ± [ 0.10574   0.15613   0.00673   0.01841  0.00000 ]


Extrinsic parameters (position of right camera wrt left camera):

Rotation vector:             om = [ -0.04009   -0.03104  -0.01371 ] ± [ 0.03216   0.04000  0.00405 ]
Translation vector:           T = [ -89.22539   5.26829  -1.35786 ] ± [ 2.48430   1.49136  8.63293 ]
```

(b) Intrinsic and Extrinsic Parameter for Stereo Camera after Optimization

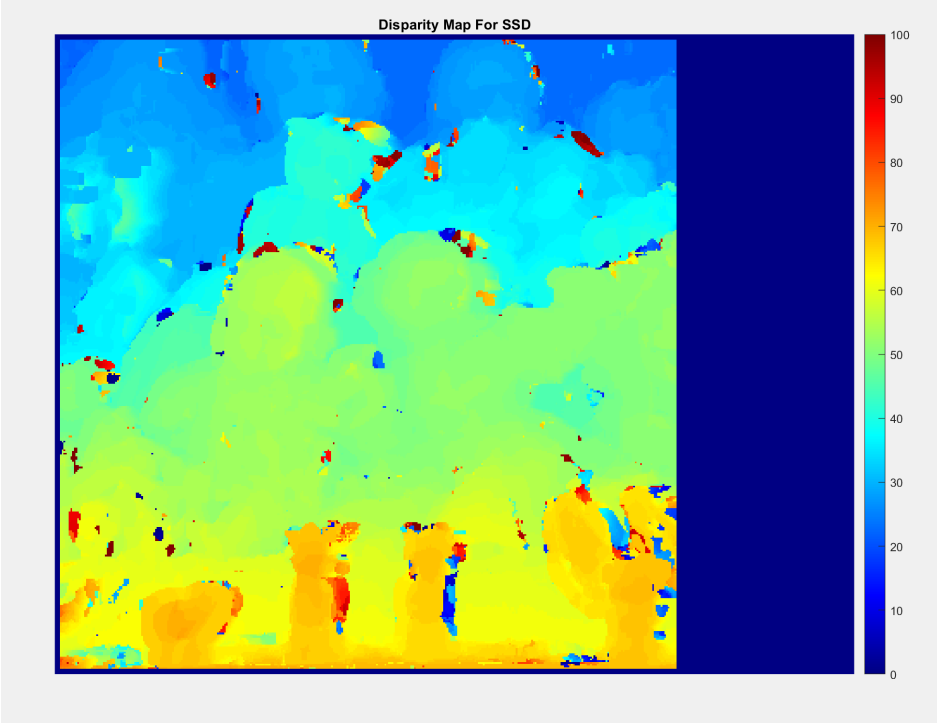
Figure 3: Intrinsic and Extrinsic Parameter for Stereo Camera

Problem 2

In this problem, we need to find the disparity mapping between two stereo images. We need to find the disparity map using Sum of squared differences (SSD), Cross-correlation (CC) and Normalized cross-correlation (NCC) algorithm. I have taken the window size as 7 as it gives optimum disparity. Based on the result and calculation time I think NCC performs well eventhough it takes much time. The disparity map along with the errors are given below:

```
Time taken for computing disparity map is 29.72 sec.

 Resul for SSD

 Max error for SSD 73
 Min error for SSD -100
 Mean error for SSD 10
 Standard Deviation of error for SSD 22
 ************************
```

(a) Errors with Ground truth using SSD

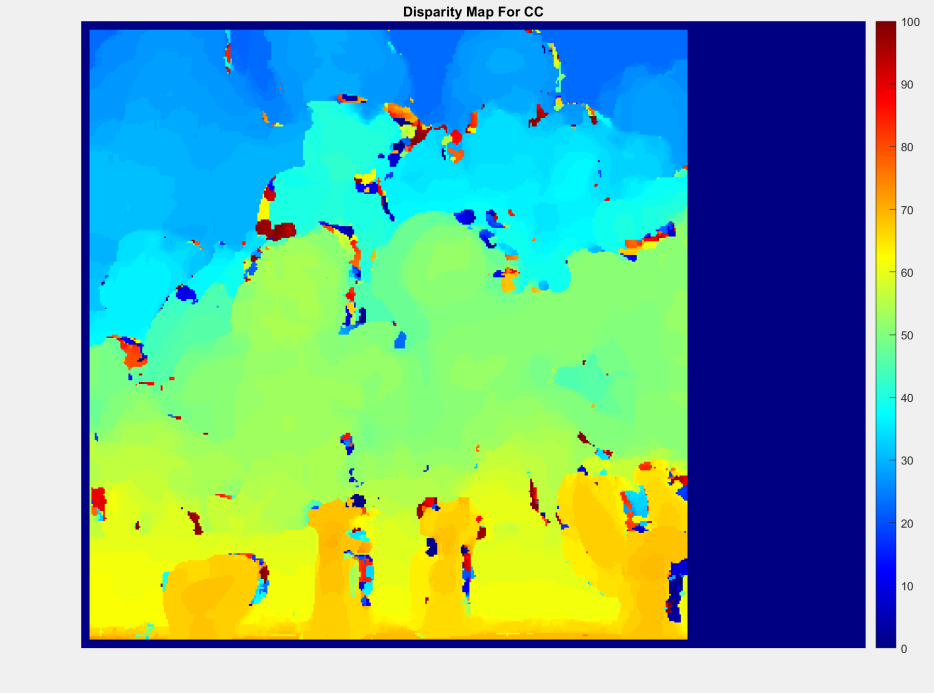


(b) Disparity Mapping

Figure 4: Disparity maps for SSD

```
Time taken for computing disparity map is 28.34 sec.

 Resul for CC

 Max error for CC 73
 Min error for CC -100
 Mean error for CC 12
 Standard Deviation of error for CC 36
 ************************
```

(a) Errors with Ground truth using CC

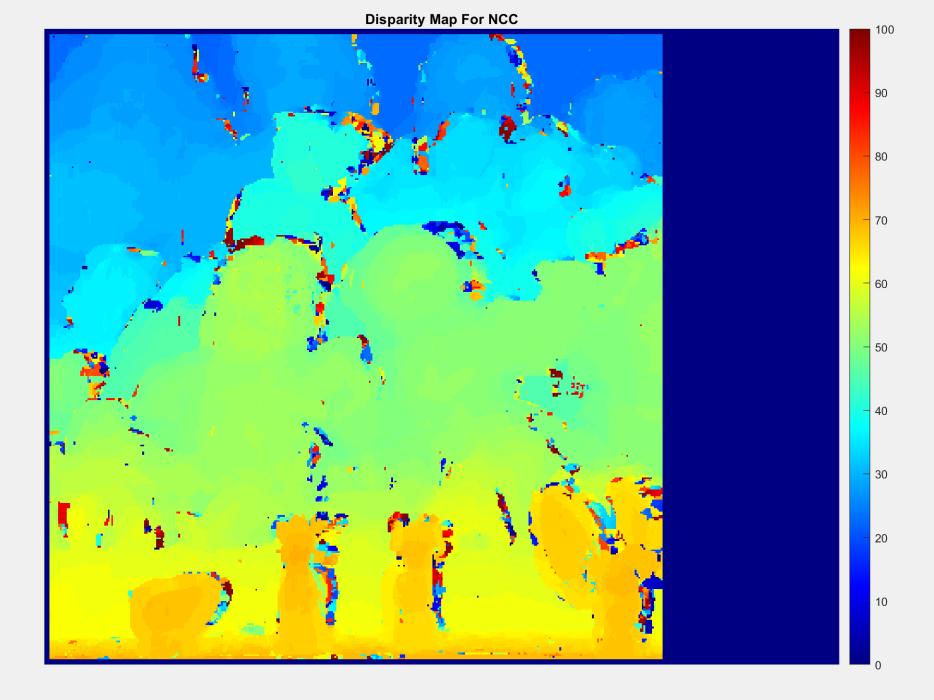


(b) Disparity Mapping

Figure 5: Disparity maps for CC

```
Time taken for computing disparity map is 81.35 sec.

 Resul for NCC

 Max error for NCC 73
 Min error for NCC -99
 Mean error for NCC 10
 Standard Deviation of error for CC 22
 ************************
```

(a) Errors with Ground truth using NCC



(b) Disparity Mapping

Figure 6: Disparity maps for NCC

Problem 3

In this particular problem we need to first rectify the two images, then compute the correspondence between them, and finally produce a depth map and a 3D point cloud for the scene. Firstly, we need to compute the camera projection matrix. Based on the data that we got from problem 1, first we need to find the camera matrix for left and right camera. We know for the pinhole camera model-

$$P = K[R|T]$$

where, p is the camera matrix, K is the the instrinsic parameter, R is the rotation matrix and T is the transalation matrix. i have found out the camera parameter for both the camera but unable to rectify the stereo images.

Problem 4

In this probelm, we are given a dataset firstly, we need to draw the best fitted line using least square approach for both dataset. then using ransac based algorithm, we need to remove the outlier. In solving the ransac based algorithm, I have used a threshold to get closed point for fitted line. then I used this parameter for comparing highest number of closest points for the fitted line. Based on this approach, the outliers are removed and only the points less than the threshold remains in the fitted scenerio. For this problem, I have defined two function **least_square.m** and **ransac.m** for doing the least square and ransaca based algorithm respectively.

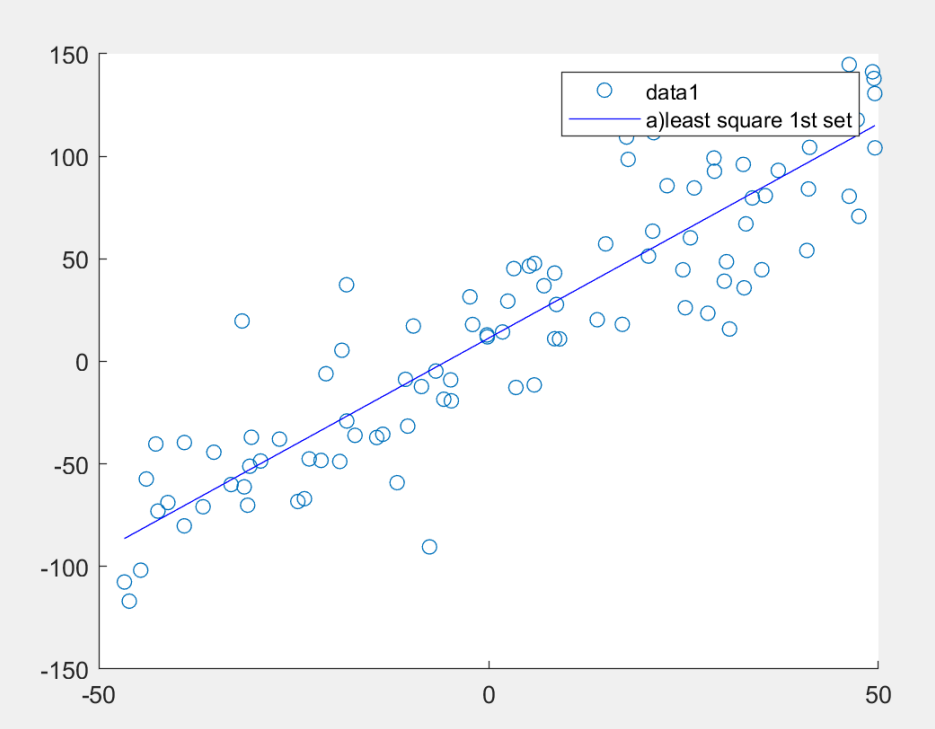


Figure 7: line fitting of first set of measurements using a least squares approach

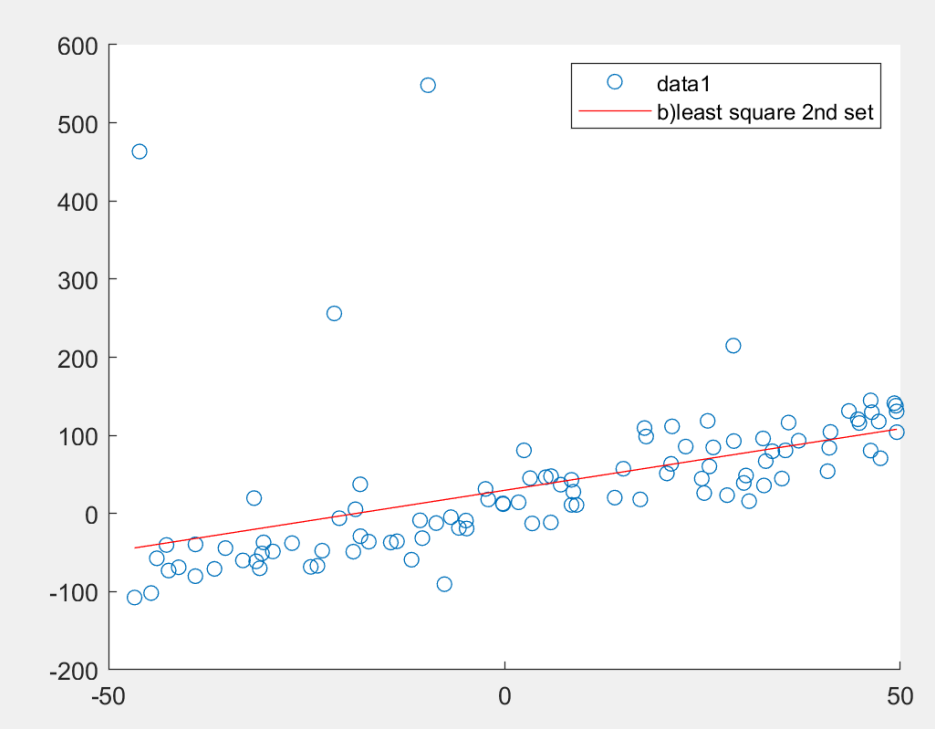


Figure 8: line fitting of second set of measurements using a least squares approach

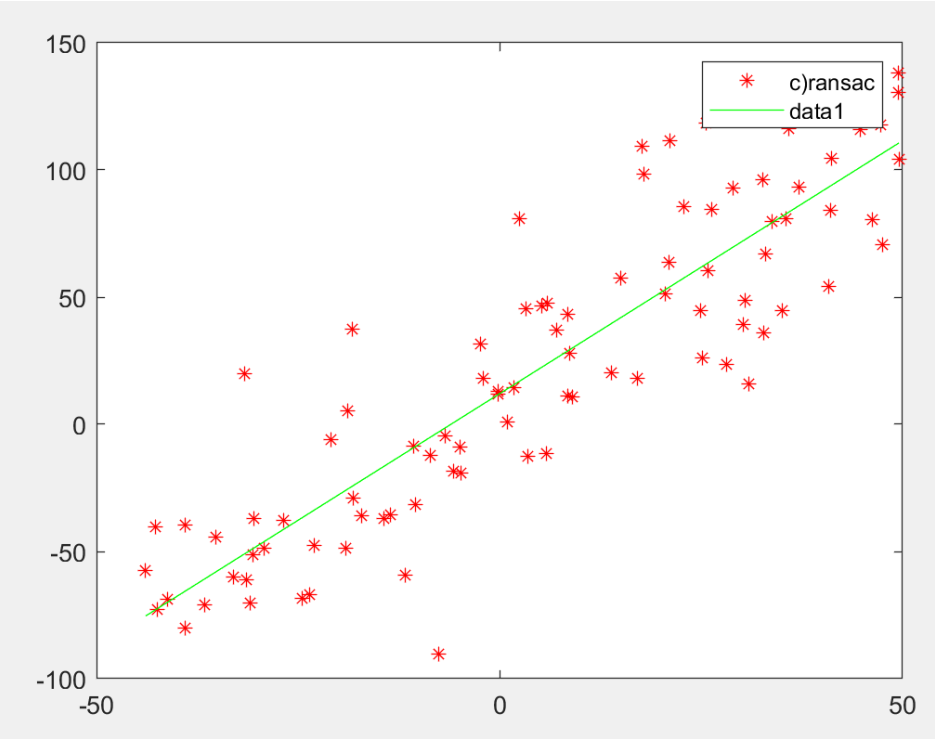


Figure 9: RANSAC-based algorithm to get rid of the outliers

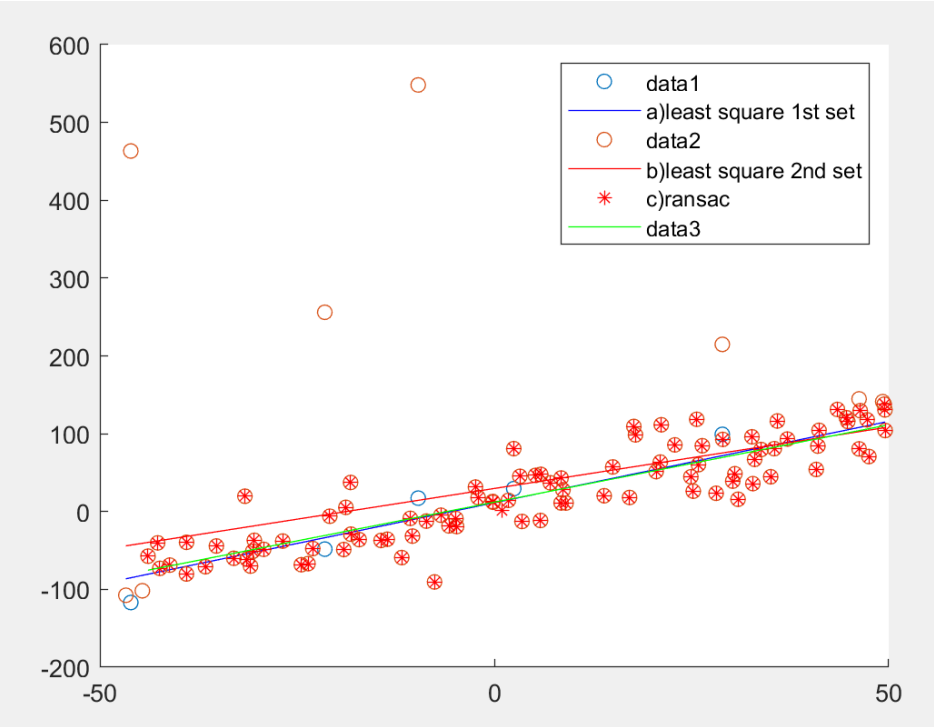


Figure 10: All the line fit in the same window