Aditya Das (axd5763)

1001675762

Assignment 4

Problem 1

1a) This problem has 2 functions:

1. function [Kl,Tl,Kr,Tr] = compute_stereo_calib(P,pl,pr) - This calculates the intrinsic and extrinsic properties of the camera

2. function [Pl] = compute_cam(P,pl) - This calculates the camera matrix.

To get the desired output, run the program like this:

[Kl,Tl,Kr,Tr] = compute_stereo_calib(P,pl,pr).

```
Intrinsic matrix for left image

Kl =

  347.3569   -1.5114  306.2412
         0 -344.9507   58.0283
         0         0    1.0000

Intrinsic matrix for right image

Kr =

 -359.4023   -1.3201  302.5369
         0 -359.6237   39.5911
         0         0    1.0000

Extrinsic properties for left image

extL =

   -0.4857    0.8741    0.0015    2.2127
   -0.0192   -0.0089   -0.9998    0.2254
   -0.8739   -0.4857    0.0211   10.0712

Extrinsic properties for right image

extR =

   -0.6982    0.7159   -0.0070    3.4676
   -0.0253   -0.0149    0.9996    0.3300
    0.7155    0.6980    0.0285  -10.9576
```

The following result was generated.

Problem 3

After importing the images and doing the calibration, the following properties were first generated.

The intrinsic left camera matrix is

```
>> cameraParamsleft.IntrinsicMatrix'

ans =

  672.6666         0  344.5111
         0  669.4044  249.7606
         0         0    1.0000
```

The intrinsic right camera matrix is

```
>> cameraParamsright.IntrinsicMatrix'

ans =

  671.5537         0  344.5889
         0  670.1728  235.3563
         0         0    1.0000
```

The following are the translational vectors for the left camera images are

cameraParamsleft.TranslationVectors

	1	2	3
1	-116.4781	-102.6758	718.7440
2	-104.7861	87.5248	720.5647
3	-63.3901	121.2985	1.0877e...
4	-77.6462	-132.2293	919.7230
5	-182.7232	-127.2876	936.3021
6	-156.0845	-88.6734	864.2941
7	-75.8650	-67.6493	1.0171e...
8	169.5067	-176.2968	1.0249e...
9	-148.2552	-426.7943	1.4132e...
10	-138.0053	-282.5400	1.0376e...
11	-140.3237	-174.4811	821.0244
12	-109.0023	-45.0614	585.6715

The following are the translational vectors for the right camera images are

cameraParamsright.TranslationVectors

	1	2	3
1	-217.9052	-76.7860	718.3107
2	-206.3526	105.1294	719.0536
3	-172.2130	144.0746	1.0862e...
4	-176.3701	-110.9057	913.3099
5	-290.4038	-105.1980	938.5807
6	-265.5094	-70.5409	872.7633
7	-186.7500	-45.6175	1.0146e...
8	42.3127	-152.7690	1.0390e...
9	-267.5866	-397.6192	1.4232e...
10	-251.8621	-266.7228	1.0615e...
11	-246.1245	-158.4217	823.1256
12	-208.3726	-30.9378	587.1048

Left camera Skew

cameraParamsleft.Skew

	1	2
1	0	

Right camera Skew

cameraParamsright.Skew

	1	2
1	0	

Right Camera's focal point and principal point

FocalLength	[671.5537,670.1728]
PrincipalPoint	[344.5889,235.3563]

Left Camera's focal point and principal point

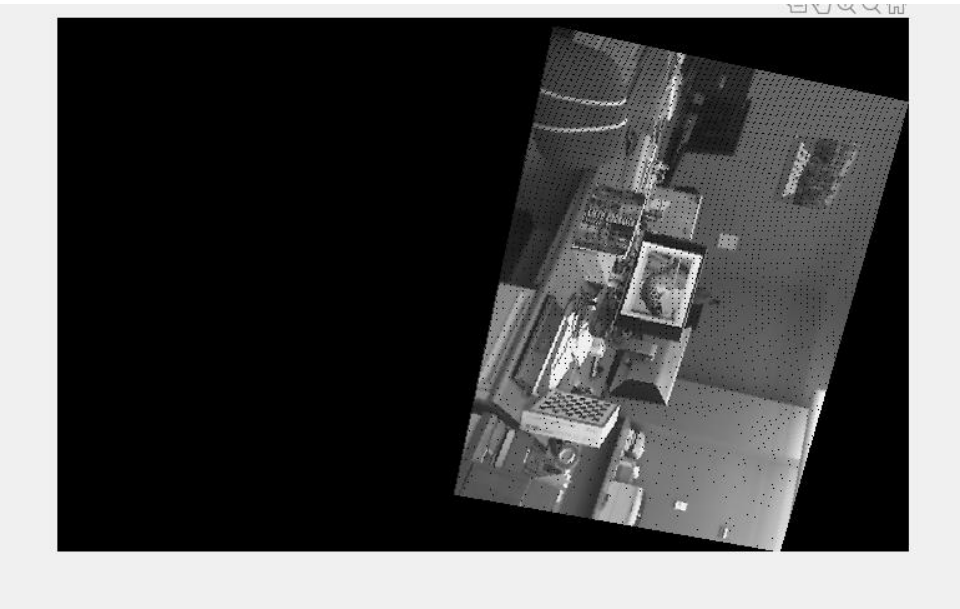
FocalLength	[672.6666,669.4044]
PrincipalPoint	[344.5111,249.7606]

The extrinsic rotational matrixes (both left and right) are attached with the report as an excel sheet(Rotational matrix.xlsx)

Problem 4a(incomplete)

This problem has one script and one function.

We need to have the images , and the CameraParamets stored in the form of cameraParamsright and cameraParamsleft to get output.



This is the left image.

Problem 5

This script has 5 functions:

1.function [] = compute_meanline(xs,n_y1) :- This is a variation of the line fitting least squares approach without usingSVD.

2.function [] = compute_meanlineSVD(xs,n_y1):- This is a variation of the line fitting least squares approach using the moore penrose pseudo inverse method and SVD.

3.function[xran1,xran2,yran1,yran2] = ransac(xs,n_y1):- This is a ransac based algorithm to eliminate the outliers

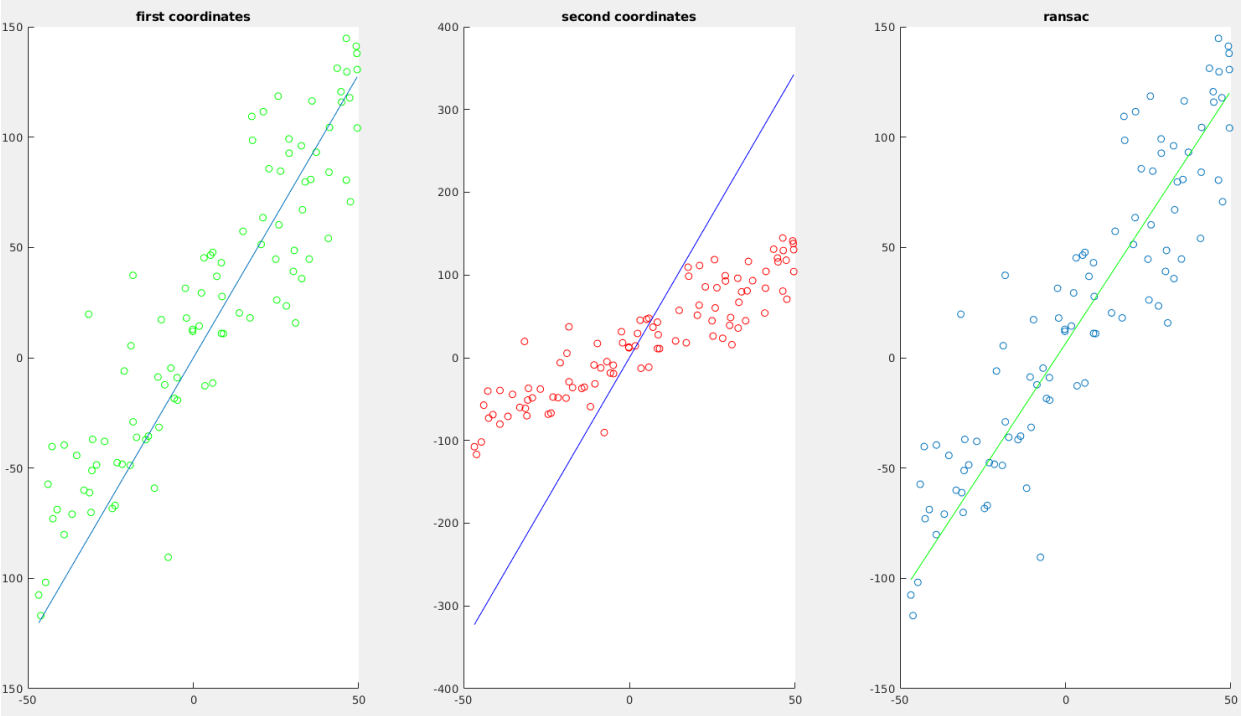
4.function d = point_to_line(pt, v1, v2):- This is a function used in the ransac method to calculate the distance from a line to a point.

5. function [] = problem(xs,n_y1,n_y2): - this calls all the other functions and plots them.

Note

The first function is commented. If the evaluator wants to check out the results of the first function, it needs to be uncommented and the rest of the function in problem commented.

Upon running the script, the following result was generated.



5a and 5b

```
line parameters of the first coordinates are

a =

   -0.9318


b =

    0.3623


c =

    0.0238

line parameters of the second coordinates are

a =

   -0.9897


b =

    0.1433


c =

   8.4650e-04
```

5c)

The Ransac based algorithm chooses the inliers based on the distance from the margin of the best fit line. The line with the most inliers is chosen as the best fit. The threshold distance is an average of the coordinates of the points.

```
line parameters of the ransac algorithm is

xran1 =

    32


xran2 =

     2


yran1 =

    92


yran2 =

    23
```