UNIVERSITY OF TEXAS AT ARLINGTON

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COMPUTER VISION

ASSIGNMENT 5

Nudrat Nawal Saber

1001733394

Problem 1 (25 points) The images ‘checkerboard[1-12][l—r].png’ in ‘checkerboard.zip’ were taken using a stereo system. Each square on the checkerboard is of dimensions (25.4mm × 25.4mm).Use the MATLAB camera calibration toolbox (http://www.vision.caltech.edu/bouguetj/calib doc/) to extract the intrinsic parameters for each camera and the extrinsic parameters giving the transformation for points in the right camera frame to those in the left camera frame. Submit the obtained camera parameters in the report.

Solution:

To get the intrinsic and extrinsic parameters the steps were followed:

i. I took the left camera picture in one folder ,and right camera pictures in other folder.I used the camera Calibrator App and the load the 12 left camera images into the application with 25.4mm.For left camera I save as leftcameraparams.I saved the session into

ii. LeftcalibrationSession.mat .

iii. I did the same for right camera images . I saved the session into rightcalibrationSession.mat .

iv. I Ran the script Problem_1.m to find the intrinsic and extrinsic parameters which can be seen below.

```
>> Problem_1
LEFT CAMERA INTRINSIC
  672.6666         0  344.5111
         0  669.4044  249.7606
         0         0    1.0000
LEFT CAMERA EXTRINSIC
    0.8656   -0.1089    0.1914  -95.2544
    0.0338    0.8530    0.1687 -117.9055
   -0.1796   -0.1190    0.7422  928.9005
RIGHT CAMERA INTRINSIC
  671.5537         0  344.5889
         0  670.1728  235.3563
         0         0    1.0000
RIGHT CAMERA EXTRINSIC
    0.8647   -0.1116    0.2206 -203.9281
    0.0424    0.8522    0.1401  -97.1929
   -0.2024   -0.0917    0.7411  933.0619
```

Problem 2 (25 points) Figure 1 shows two images of a scene taken from the left camera (viewL.png) and right camera (viewR.png) of a stereo system. The images have been rectified and are free from radial distortion. Compare the results of following algorithms for computing correspondences and generating a disparity map:

- Sum of squared differences (SSD)
- Cross-correlation (CC)
- Normalized cross-correlation (NCC)

Note that some of these values must be minimized whereas others must be maximized. You will also need to experiment with window sizes and decide on a method to disambiguate in case of ties. Report these decisions as part of your answer. You can use MATLAB functions to convert color images to gray scale. The ground truth for the disparity (right - left) is provided in the file 'disparity.mat'. Write a MATLAB function compute corrs that reads in the images and computes the disparity map using each of the three methods. For each method determine the mean, min, max and standard deviation of the error values using the ground truth. Also report the running times of each method implemented. Using the above criteria, report which method is the best\

Solution:

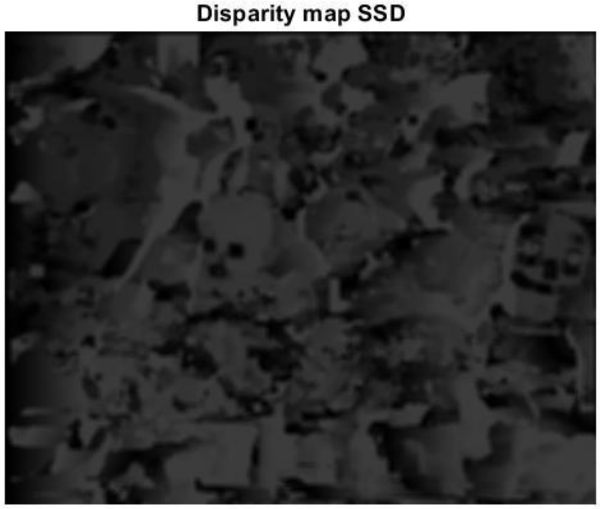
For the implementation I selected window size 5 and chose the threshold 60 .Images are padded based on the size of the window.A patch is created for every pixel .and so as a strip from right side alse created.The search is done based on the threshold and we got the best disparity.

```
>> problem_2
SSD
SSDtime is: 00:00:57in HH:MM:SS
Error: 2741682.64
Mean Squared Error: 447.6884
CC
CCtime is: 00:01:26in HH:MM:SS
Error: 3034772.52
Mean Squared Error: 512.9986
NCC
NCCtime is: 00:01:25in HH:MM:SS
Error: 3034772.52
Mean Squared Error: 512.9986
```

Disparity map CC

Disparity map NCC

Disparity map SSD



According to the figures ,it can be decided that SSD gives the best result.

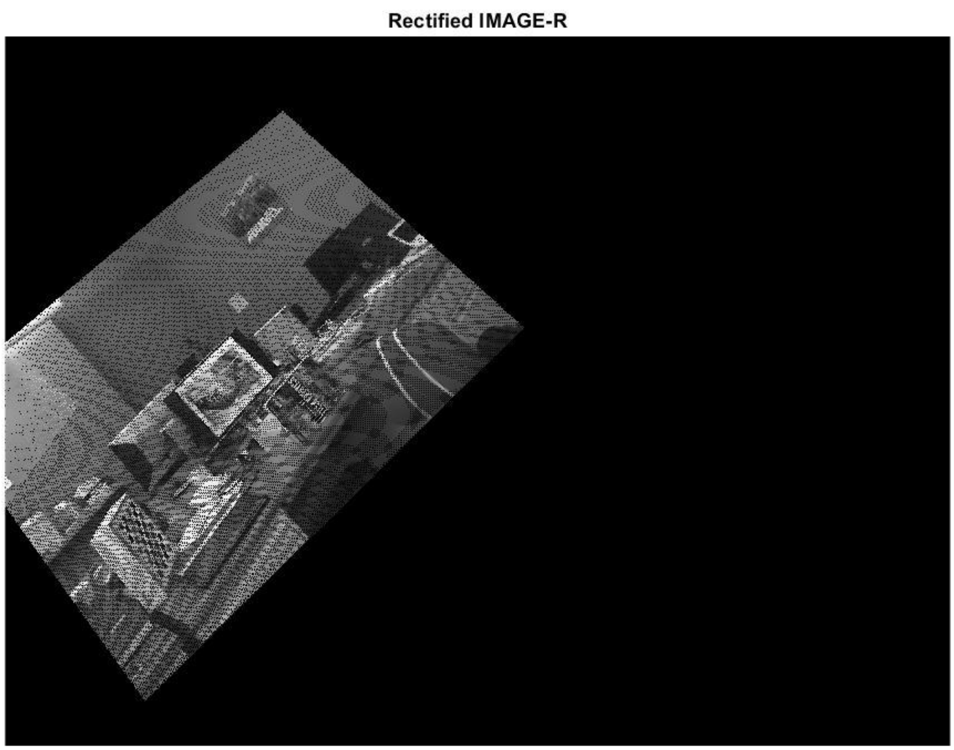
Problem 3 (15 + 10 = 25 points) Images ‘officeL.png’ and ‘officeR.png’, shown in Figure 2, were taken from the stereo pair used for the previous problem. In this problem, you are required to first rectify the two images, then compute the correspondence between them, and finally produce a depth map and a 3D point cloud for the scene.

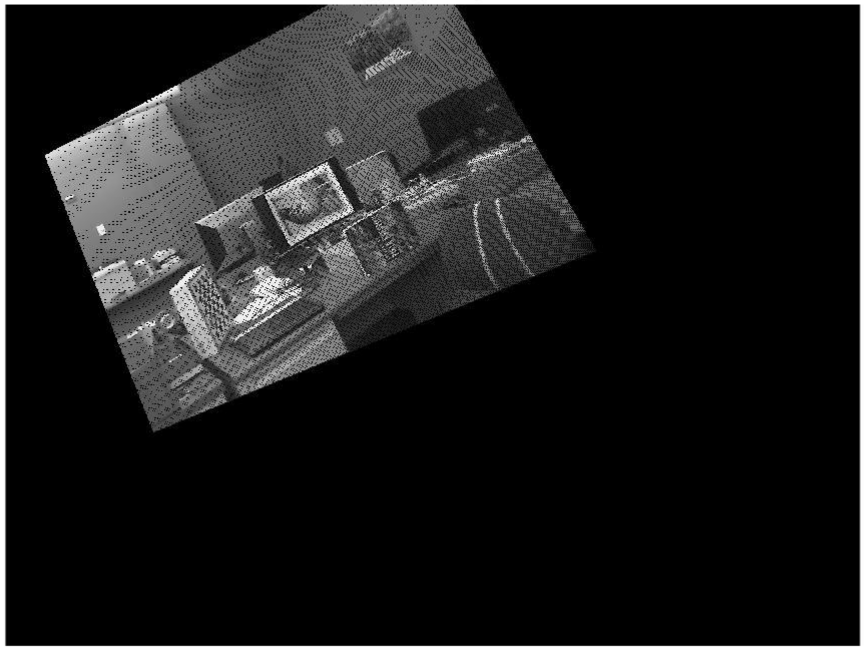
(a) Using the camera parameters obtained in Problem 1, write a MATLAB function [rectL,rectR] =rectify images(imgL,imgR,Pl,Pr) that takes in the left and right images (imgL and imgR) along with the 3 × 4 camera projection matrices Pl and Pr, and produces the rectified images rectL and rectR.

(b) Use your function compute corrs in Problem 2 to determine the correspondences between the pair of rectified images using the better of the three methods (SSD, CC, NCC). Using these correspondences along with the camera parameters, determine the 3D locations for all points in the image. (You may choose to ignore those points which do not occur in the other image.) Display the results using a 2D depth map and a 3D point cloud in MATLAB. For this part, submit a script ‘problem 3.m’ that reads in the image and camera parameters, rectifies them, and generates the point clouds.

Solution:

a)





b) partially done

Problem 4 (5 + 5 + 15 = 25 points) The file 'linefit.mat' contains three arrays:

- xs: The x-coordinates of 100 points on a line
- n y1: The measurements of the y-coordinates, these measurements have zero-mean Gaussian noise
- n y2: A second set of measurements that is similar to the n y1, but contains outliers produced due to device anomalies

(a) Fit a line through the first set of measurements using a least squares approach. Do not use the pinv function of MATLAB. Instead, compute the pseudoinverse yourself (you can use inv). Report the line parameters.

(b) Repeat part (a) with the second set of measurements.

(c) Develop a RANSAC-based algorithm to get rid of the outliers. Explain how you chose the number of points and decided on points that are "close" to the estimated line. Report the line parameters.

Solution:

Least square method :

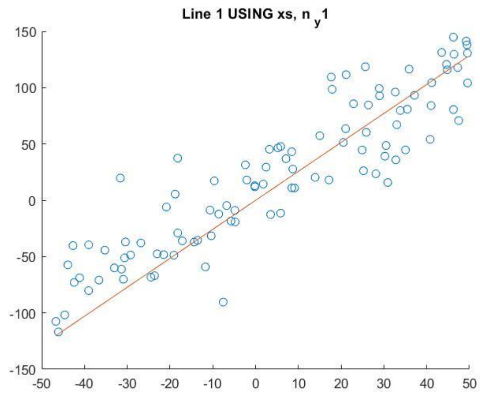
Let us assume that the given points of data are (x_1, y_1), (x_2, y_2), …, (x_n, y_n) in which all x's are independent variables, while all y's are dependent ones. Also, suppose that f(x) be the fitting curve and d represents error or deviation from each given point. The least-squares explain that the curve that best fits is represented by the property that the sum of squares of all the deviations from given values must be minimum.

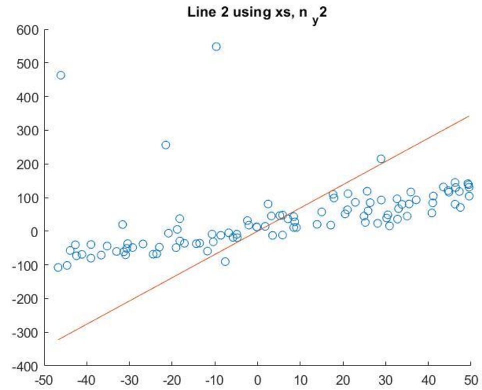
Custom Algorithm for removing outliers :

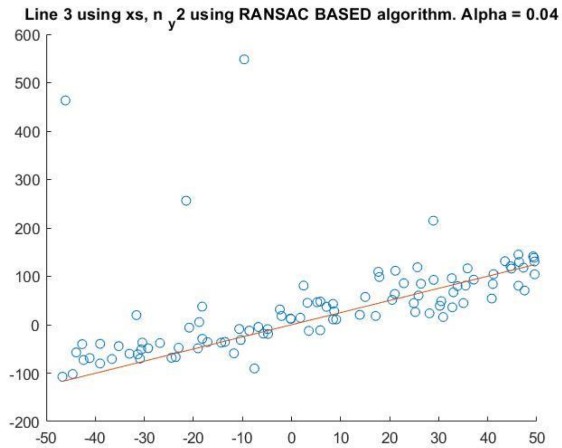
A self learning algorithm is used ,which follows neural networks. The best fit is calculated using least squared method ,and also the distance vector is calculated. The inliers are taken based on mean of d and standard deviation of d.and the best fit is computed. We repeat it until there is no change in the inliers for couple of continuous loops.

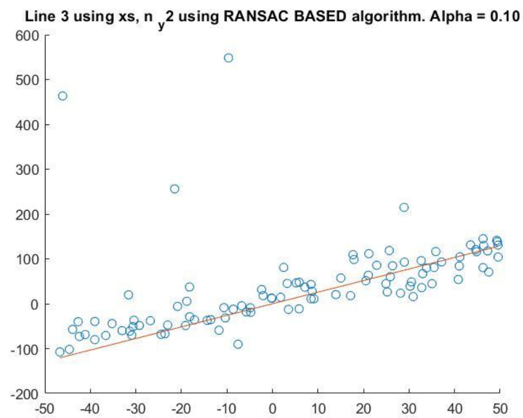
```
>> problem_4

Line1 =

   -0.9318    0.3623    0.0238

-----------------------------------

Line2 =

   -0.9897    0.1433    0.0008

###############
Outliers:
   28.9029  214.8139
  -18.2167   37.3728
    2.4637   80.8364
  -21.4892  256.0695
   -7.5773  -90.4382
  -46.0816  463.2938
  -31.6157   19.7157
   -9.6499  548.2151
```

```
Line3 =

   -0.9285    0.3704    0.0260

####################
Outliers:
   28.9029  214.8139
  -21.4892  256.0695
  -46.0816  463.2938
  -31.6157   19.7157
   -9.6499  548.2151


Line4 =

   -0.9323    0.3611    0.0219

###################
```









References:

1.https://www.youtube.com/watch?v=AGH19eJWunk

2.https://www.mathworks.com/help/vision/ug/camera-calibration.html

3.https://www.youtube.com/watch?v=GpU1Vx-b3VA&t=288s

4.https://www.youtube.com/watch?v=QqRBNbfwPII

5.https://drive.google.com/file/d/1_s2nZM6JPp566EmYdtEOCpcBUBeNMk5K/view

6.https://www.cs.auckland.ac.nz/~rklette/CCV-CIMAT/pdfs/B14-CameraCalibration.pdf

7.http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/FUSIELLO2/node5.html

8. http://www.sci.utah.edu/~gerig/CS6320-S2012/Materials/CS6320-CV-F2012-Rectification.pdf

9. http://www.cim.mcgill.ca/~langer/558/19-cameracalibration.pdf

10.http://www.sci.utah.edu/~gerig/CS6320-S2012/Materials/CS6320-CV-F2012-Rectification.pdf

11.https://link.springer.com/article/10.1007/s11042-018-6475-6

12.https://www.youtube.com/watch?v=zIeX8bqKSBs